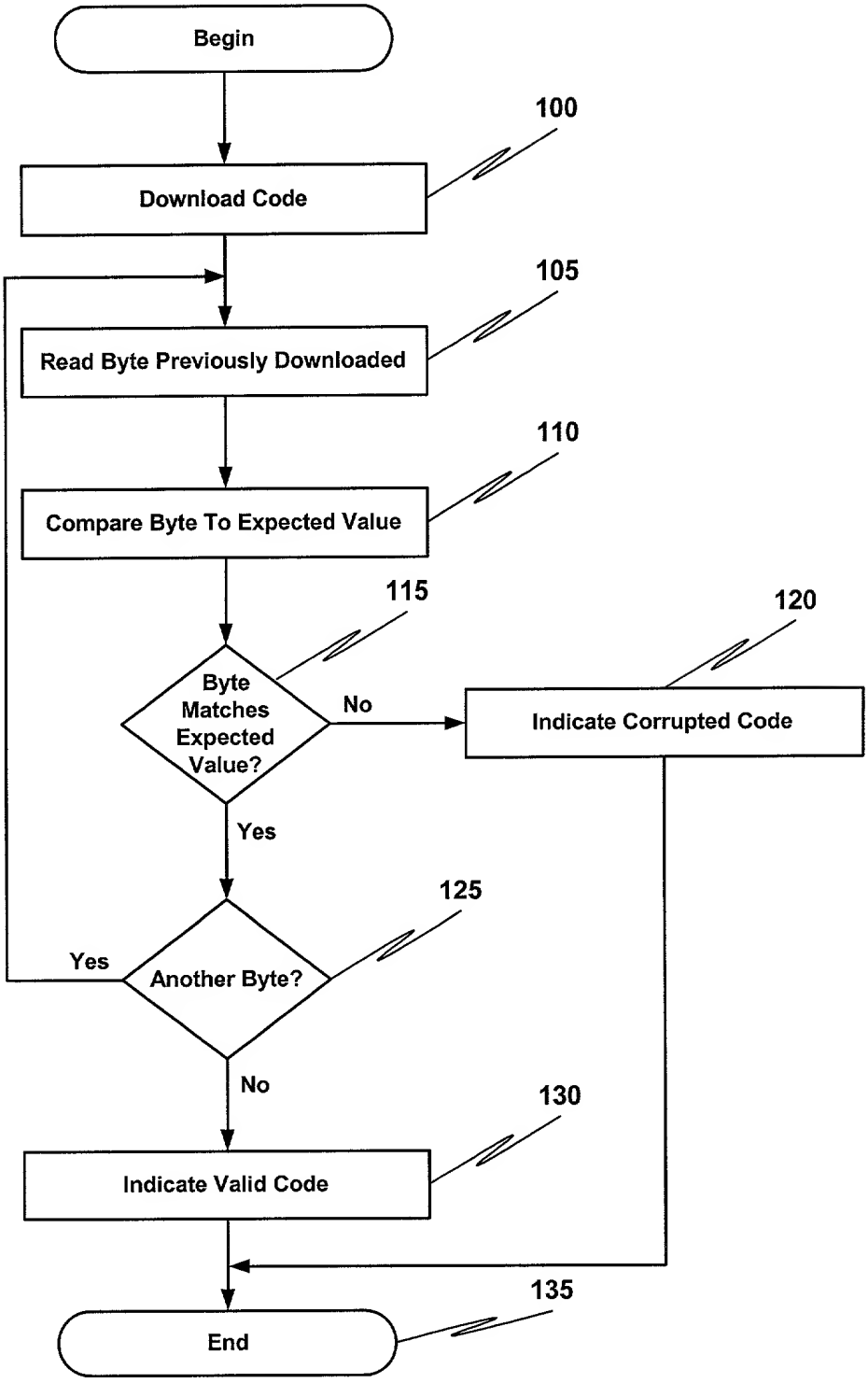


FIG. 1 - Prior Art

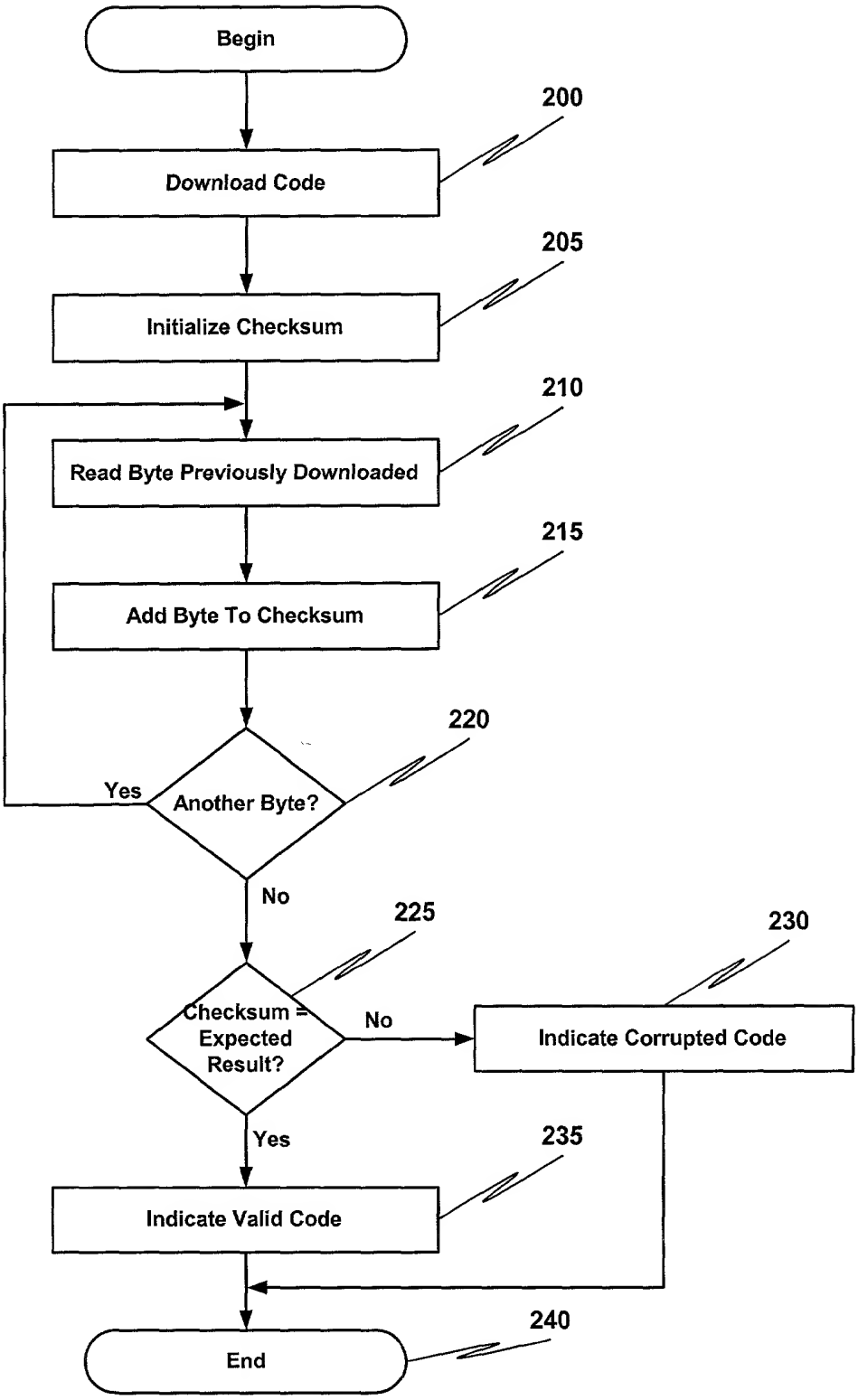


FIG. 2 - Prior Art

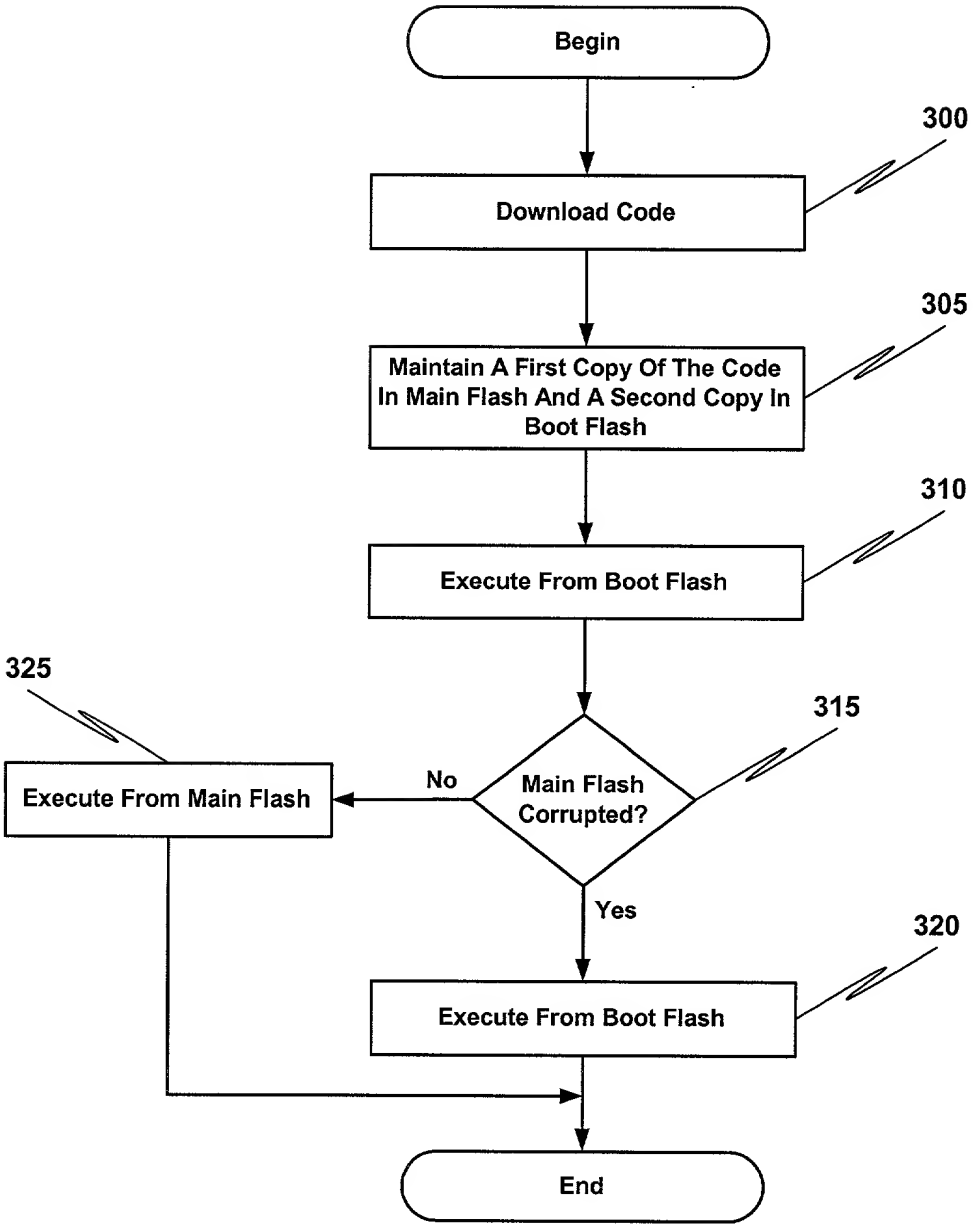


FIG. 3 - Prior Art

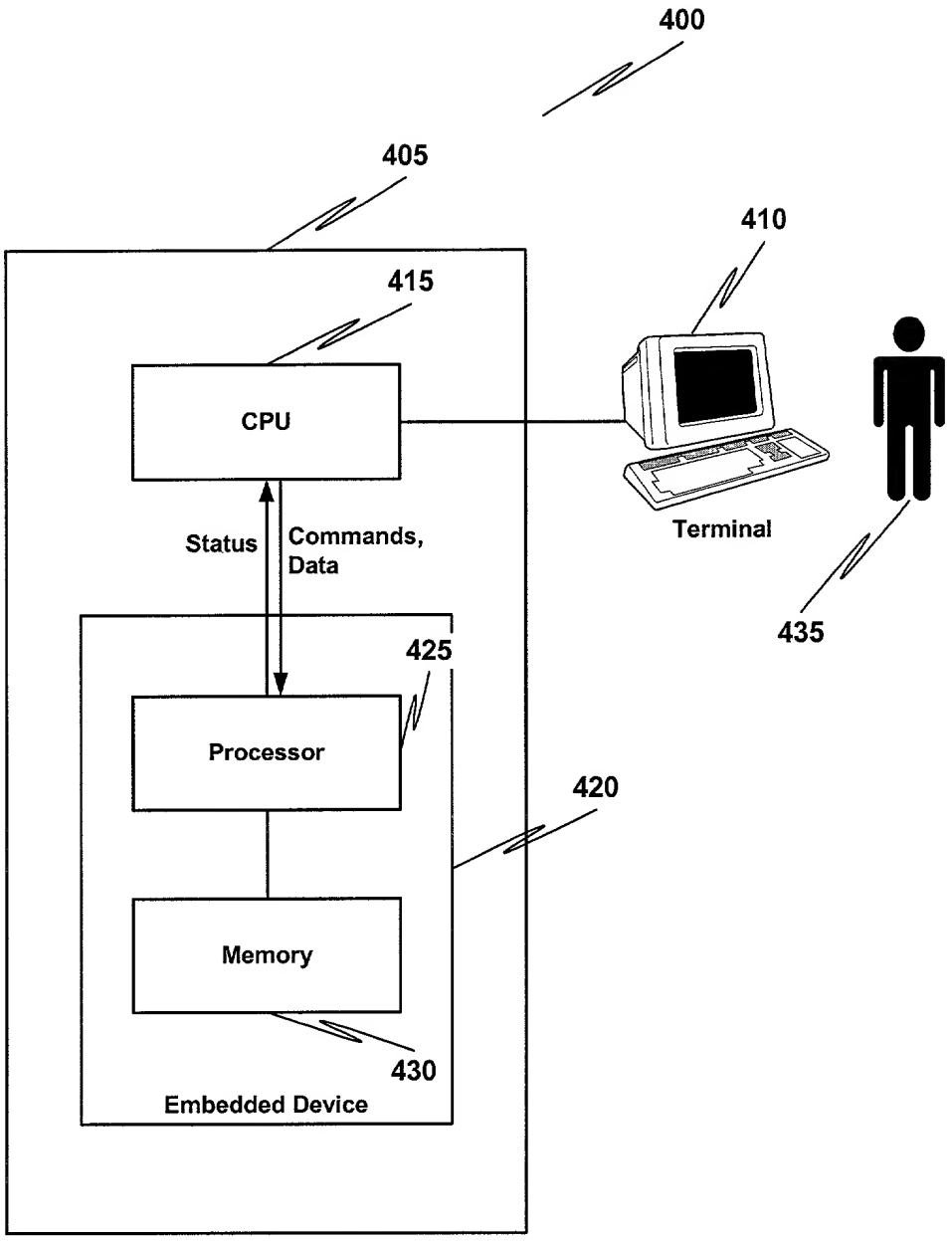

400
405
410
415
CPU
Status
Commands,
Data
Terminal
435
425
Processor
420
Memory
430
Embedded Device

FIG. 4

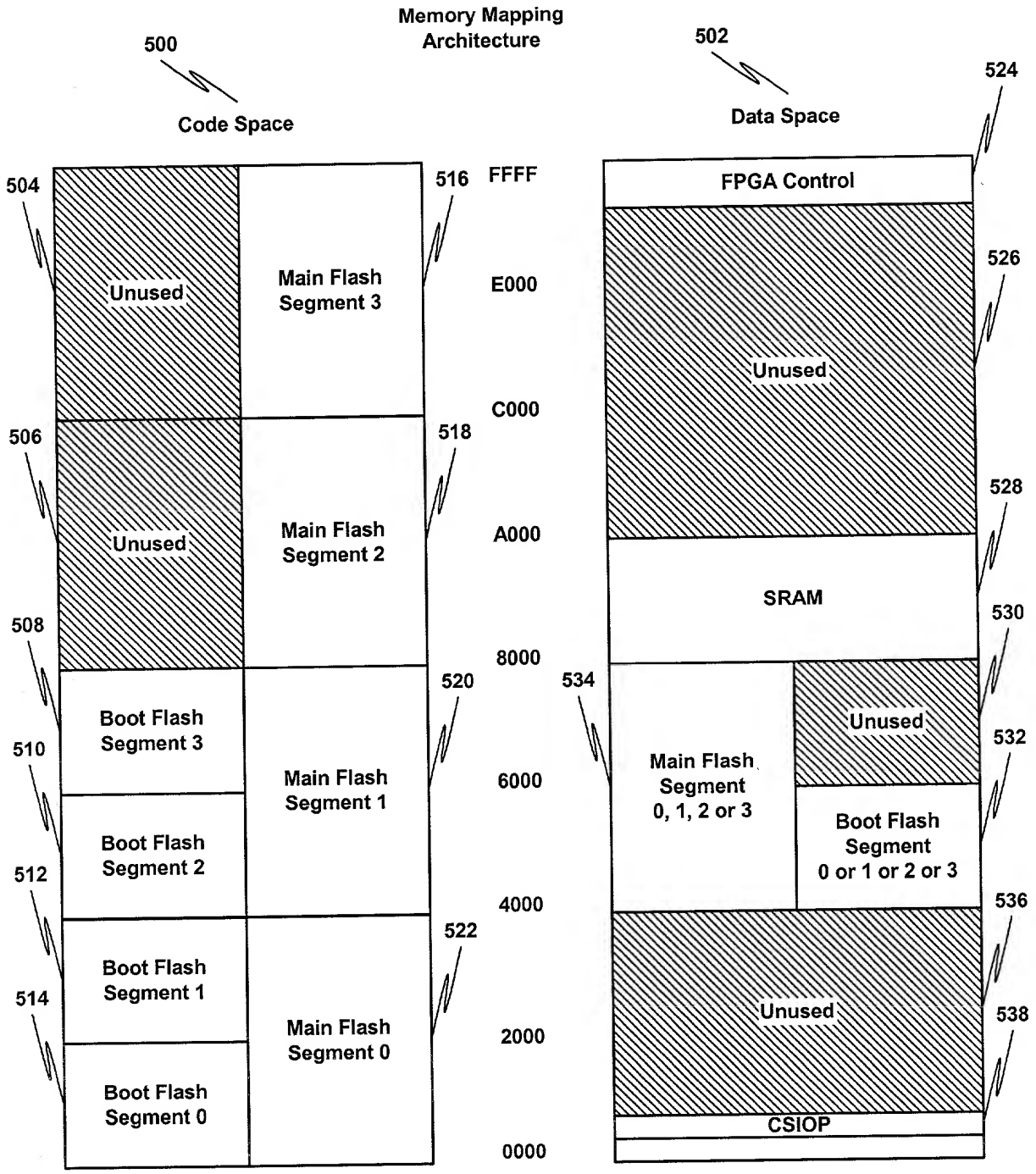
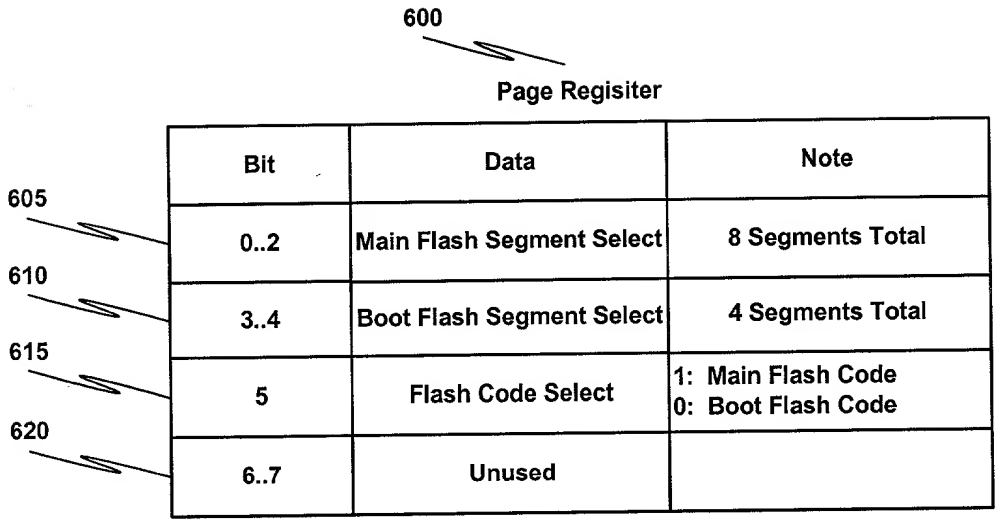
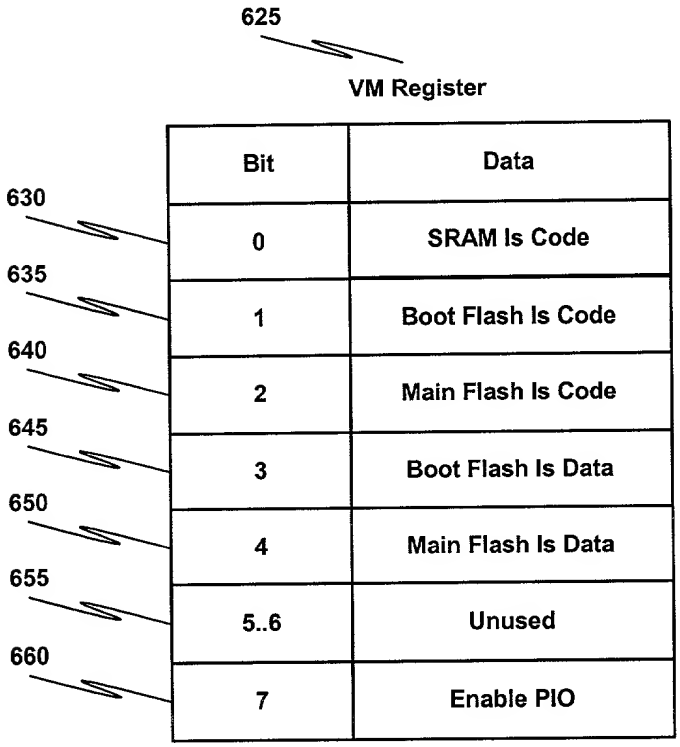

Memory Mapping Architecture
500
Code Space
502
Data Space
504
Unused
Main Flash Segment 3
516
FFFF
E000
506
Unused
Main Flash Segment 2
518
C000
A000
508
Boot Flash Segment 3
Main Flash Segment 1
520
8000
510
Boot Flash Segment 2
6000
512
Boot Flash Segment 1
Main Flash Segment 0
522
4000
514
Boot Flash Segment 0
2000
0000
524
FPGA Control
526
Unused
528
SRAM
530
534
Main Flash Segment 0, 1, 2 or 3
Unused
532
Boot Flash Segment 0 or 1 or 2 or 3
536
Unused
538
CSIOP

FIG. 5



Bit	Data	Note
0..2	Main Flash Segment Select	8 Segments Total
3..4	Boot Flash Segment Select	4 Segments Total
5	Flash Code Select	1: Main Flash Code 0: Boot Flash Code
6..7	Unused	

FIG. 6A



Bit	Data
0	SRAM Is Code
1	Boot Flash Is Code
2	Main Flash Is Code
3	Boot Flash Is Data
4	Main Flash Is Data
5..6	Unused
7	Enable PIO

FIG. 6B

Main Flash Segment 0 = (Device Space=Code) & (Flash Code Select=Main) & (address in 0000..3FFF)
OR (Device Space=Data) & (Main Flash Segment Select=0) & (address in 4000..7FFF))

Main Flash Segment 1 = (Device Space=Code) & (Flash Code Select=Main) & (address in 4000..7FFF)
OR (Device Space=Data) & (Main Flash Segment Select=1) & (address in 4000..7FFF))

Main Flash Segment 2 = (Device Space=Code) & (Flash Code Select=Main) & (address in 8000..BFFF)
OR (Device Space=Data) & (Main Flash Segment Select=2) & (address in 4000..7FFF))

Main Flash Segment 3 = (Device Space=Code) & (Flash Code Select=Main) & (address in C000..FFFF)
OR (Device Space=Data) & (Main Flash Segment Select=3) & (address in 4000..7FFF))

FIG. 7A

Boot Flash Segment 0 = (Device Space=Code) & (Flash Code Select=Boot) & (address in 0000..1FFF)
OR (Device Space=Data) & (Boot Flash Segment Select=0) & (address in 4000..5FFF))

Boot Flash Segment 1 = (Device Space=Code) & (Flash Code Select=Boot) & (address in 2000..3FFF)
OR (Device Space=Data) & (Boot Flash Segment Select=1) & (address in 4000..5FFF))

Boot Flash Segment 2 = (Device Space=Code) & (Flash Code Select=Boot) & (address in 4000..5FFF)
OR (Device Space=Data) & (Boot Flash Segment Select=2) & (address in 4000..5FFF))

Boot Flash Segment 3 = (Device Space=Code) & (Flash Code Select=Boot) & (address in 6000..7FFF)
OR (Device Space=Data) & (Boot Flash Segment Select=3) & (address in 4000..5FFF))

FIG. 7B

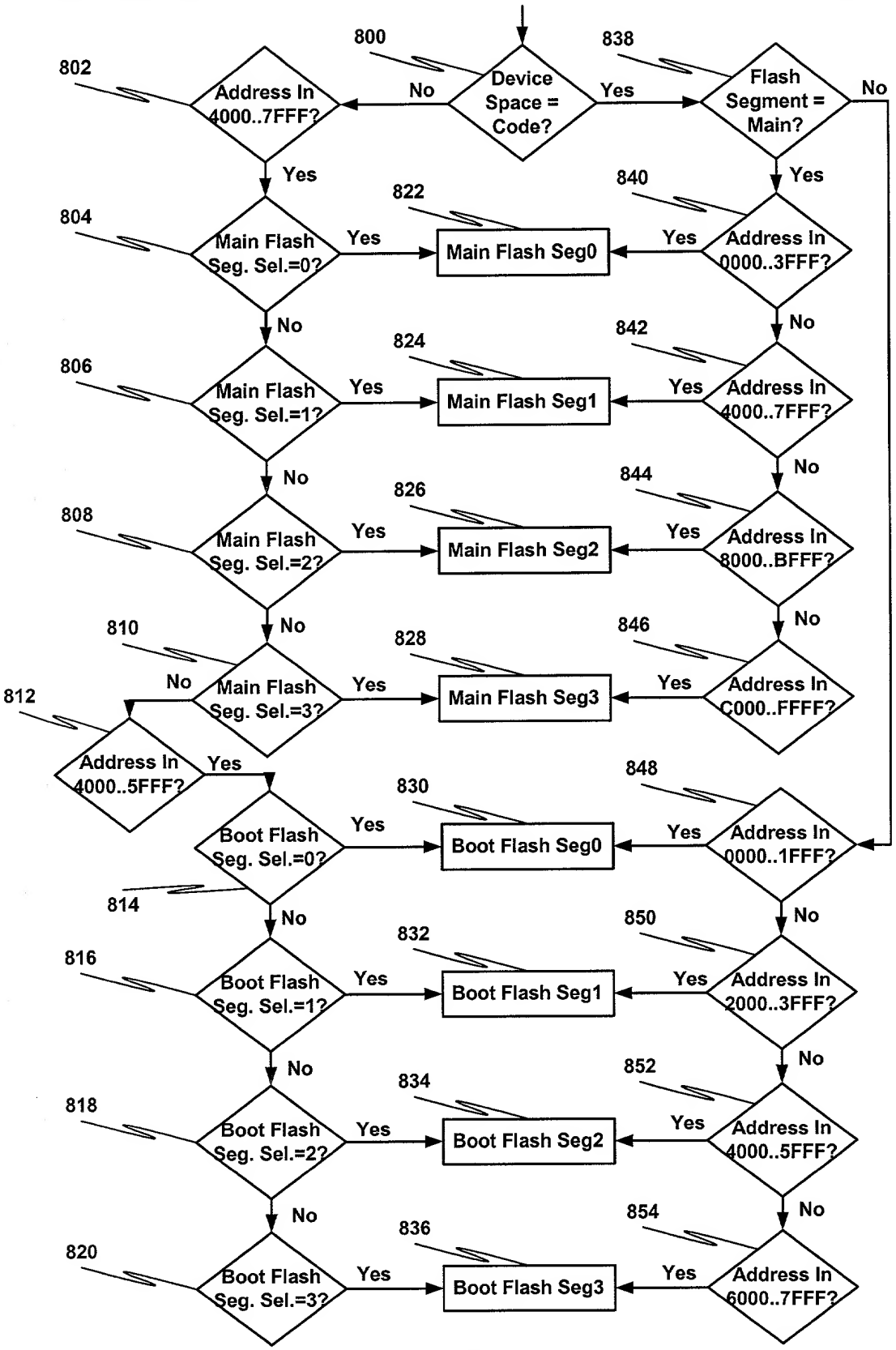

800
Device Space = Code?
No
Yes
802
Address In 4000..7FFF?
838
Flash Segment = Main?
No
Yes
804
Main Flash Seg. Sel.=0?
Yes
822
Main Flash Seg0
Yes
840
Address In 0000..3FFF?
No
806
Main Flash Seg. Sel.=1?
Yes
824
Main Flash Seg1
Yes
842
Address In 4000..7FFF?
No
808
Main Flash Seg. Sel.=2?
Yes
826
Main Flash Seg2
Yes
844
Address In 8000..BFFF?
No
810
Main Flash Seg. Sel.=3?
No
Yes
828
Main Flash Seg3
Yes
846
Address In C000..FFFF?
No
812
Address In 4000..5FFF?
Yes
830
Boot Flash Seg. Sel.=0?
Yes
Boot Flash Seg0
Yes
848
Address In 0000..1FFF?
814
No
816
Boot Flash Seg. Sel.=1?
Yes
832
Boot Flash Seg1
Yes
850
Address In 2000..3FFF?
No
818
Boot Flash Seg. Sel.=2?
Yes
834
Boot Flash Seg2
Yes
852
Address In 4000..5FFF?
No
820
Boot Flash Seg. Sel.=3?
Yes
836
Boot Flash Seg3
Yes
854
Address In 6000..7FFF?
No

FIG. 8

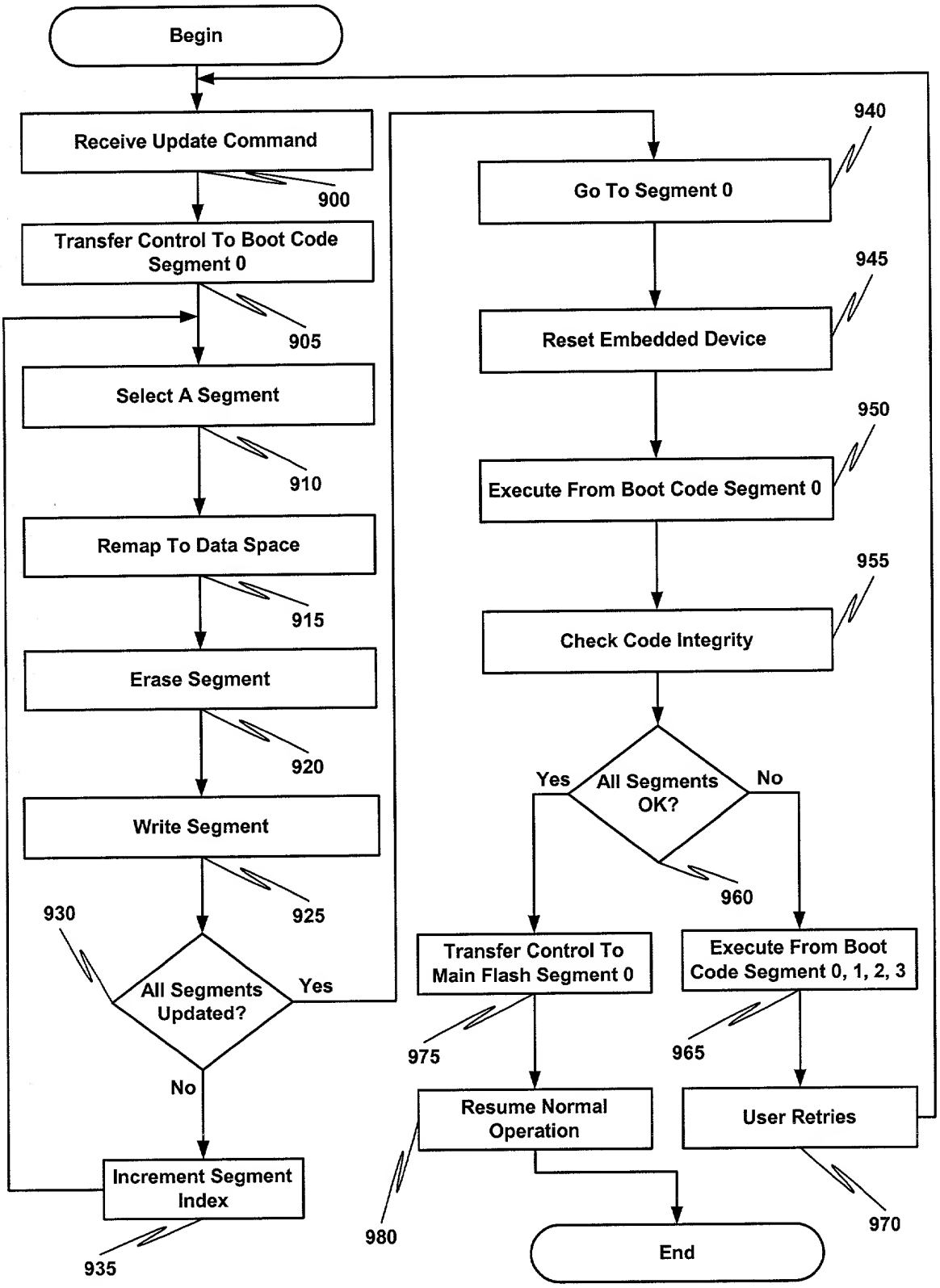

Begin
Receive Update Command
900
Transfer Control To Boot Code Segment 0
905
Select A Segment
910
Remap To Data Space
915
Erase Segment
920
Write Segment
925
930
All Segments Updated?
Yes
No
Increment Segment Index
935
940
Go To Segment 0
945
Reset Embedded Device
950
Execute From Boot Code Segment 0
955
Check Code Integrity
All Segments OK?
Yes
No
960
Transfer Control To Main Flash Segment 0
975
Execute From Boot Code Segment 0, 1, 2, 3
965
Resume Normal Operation
980
User Retries
970
End

FIG. 9

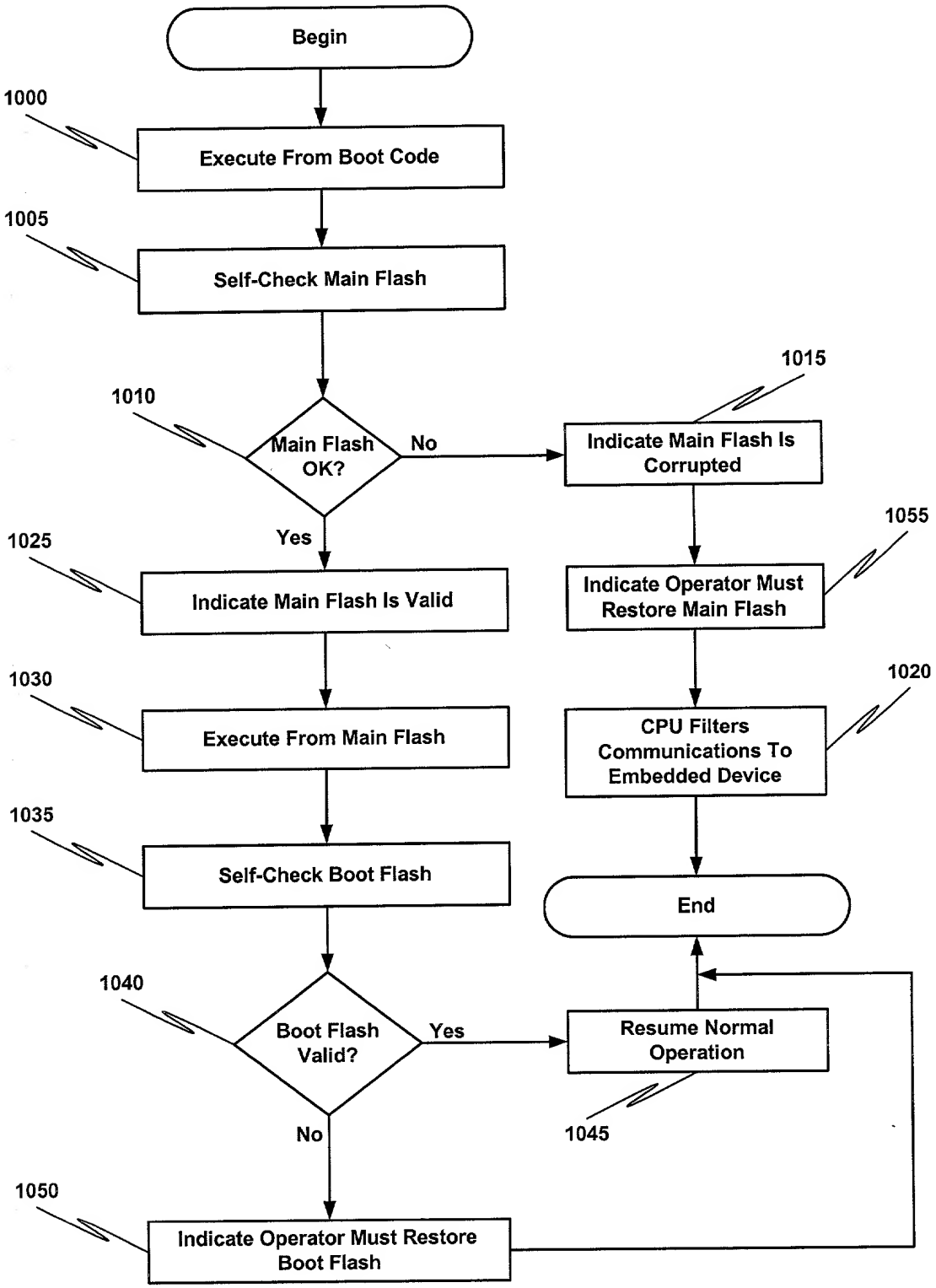

Begin
1000
Execute From Boot Code
1005
Self-Check Main Flash
1010
Main Flash OK?
No
1015
Indicate Main Flash Is Corrupted
Yes
1025
Indicate Main Flash Is Valid
1055
Indicate Operator Must Restore Main Flash
1030
Execute From Main Flash
1020
CPU Filters Communications To Embedded Device
1035
Self-Check Boot Flash
End
1040
Boot Flash Valid?
Yes
Resume Normal Operation
1045
No
1050
Indicate Operator Must Restore Boot Flash

FIG. 10

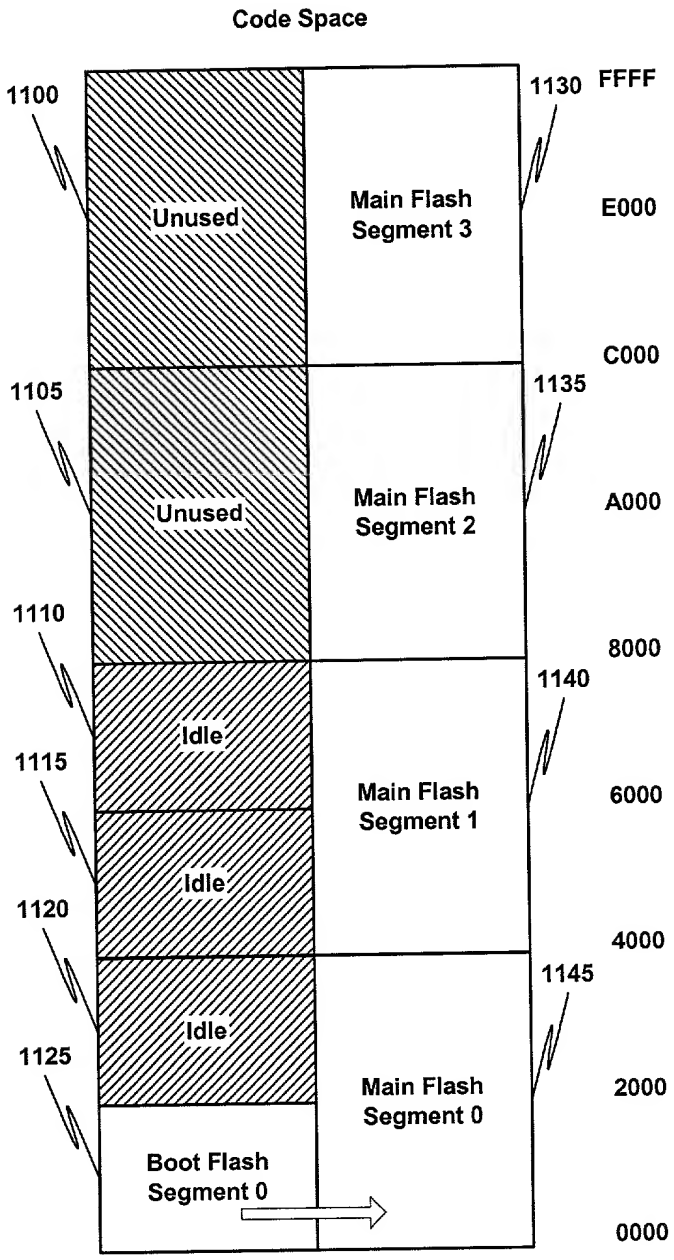

Code Space
1100
1130
FFFF
Unused
Main Flash Segment 3
E000
C000
1105
1135
Unused
Main Flash Segment 2
A000
1110
8000
1140
Idle
1115
Main Flash Segment 1
6000
Idle
1120
4000
1145
Idle
1125
Main Flash Segment 0
2000
Boot Flash Segment 0
0000

FIG. 11

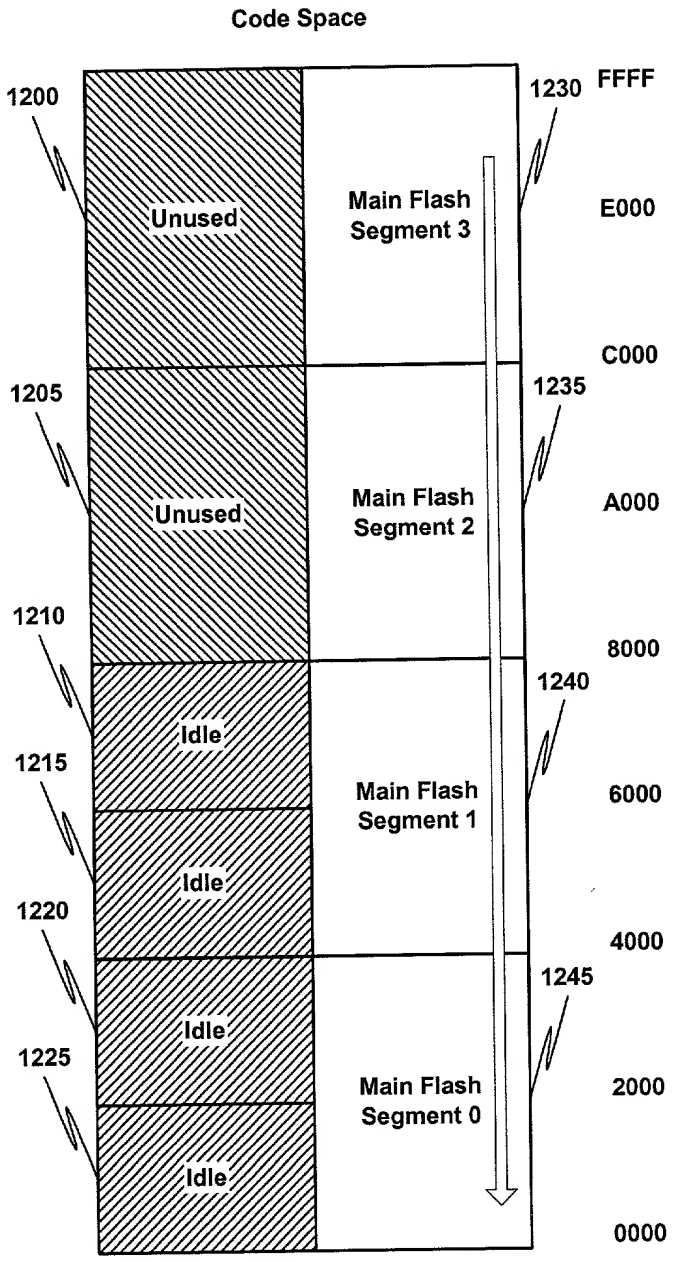

Code Space
1200
Unused
1205
Unused
1210
Idle
1215
Idle
1220
Idle
1225
Idle
Main Flash Segment 3
Main Flash Segment 2
Main Flash Segment 1
Main Flash Segment 0
1230
1235
1240
1245
FFFF
E000
C000
A000
8000
6000
4000
2000
0000

FIG. 12

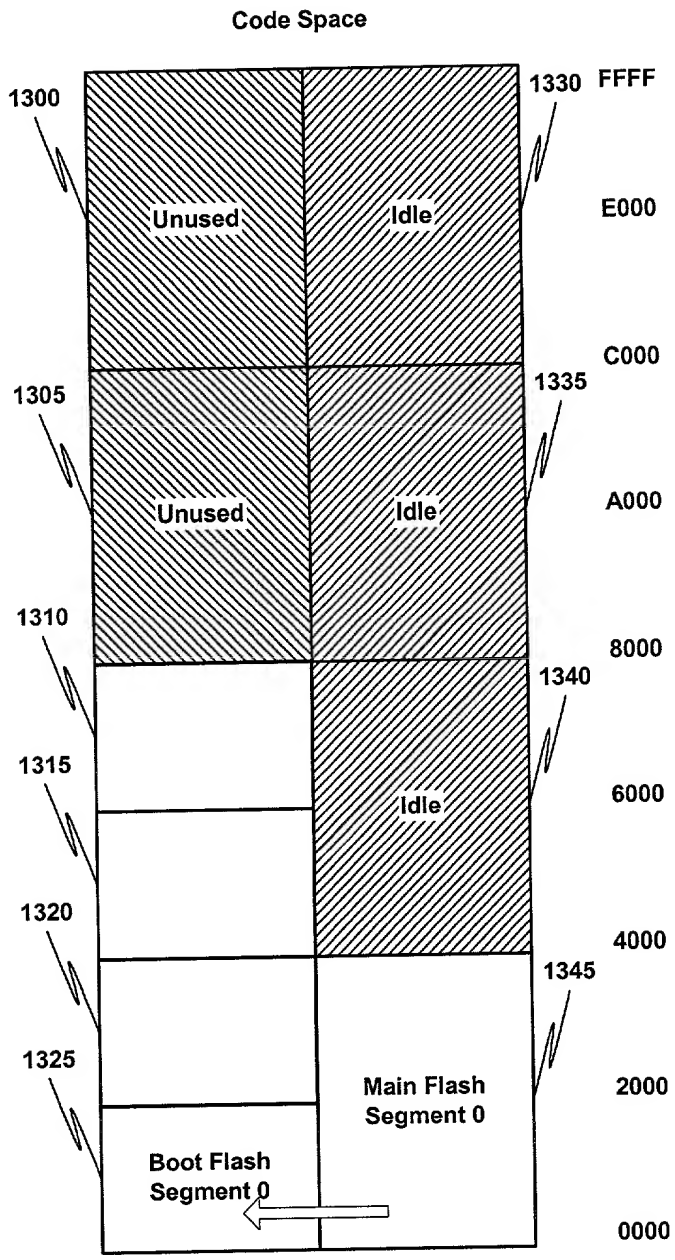

Code Space
1300
1330
FFFF
Unused
Idle
E000
C000
1305
1335
Unused
Idle
A000
1310
8000
1340
1315
Idle
6000
1320
4000
1345
1325
Main Flash
Segment 0
2000
Boot Flash
Segment 0
0000

FIG. 13

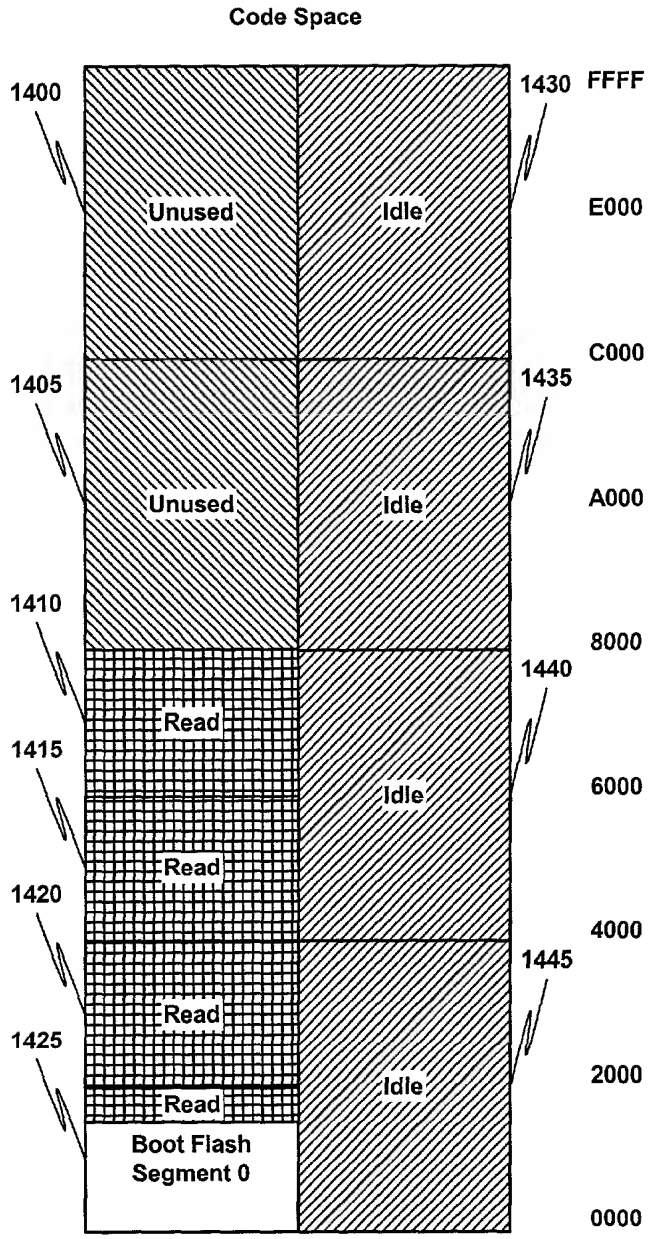

Code Space
1400
Unused
1405
Unused
1410
Read
1415
Read
1420
Read
1425
Read
Boot Flash
Segment 0
1430
Idle
1435
Idle
1440
Idle
1445
Idle
FFFF
E000
C000
A000
8000
6000
4000
2000
0000

FIG. 14

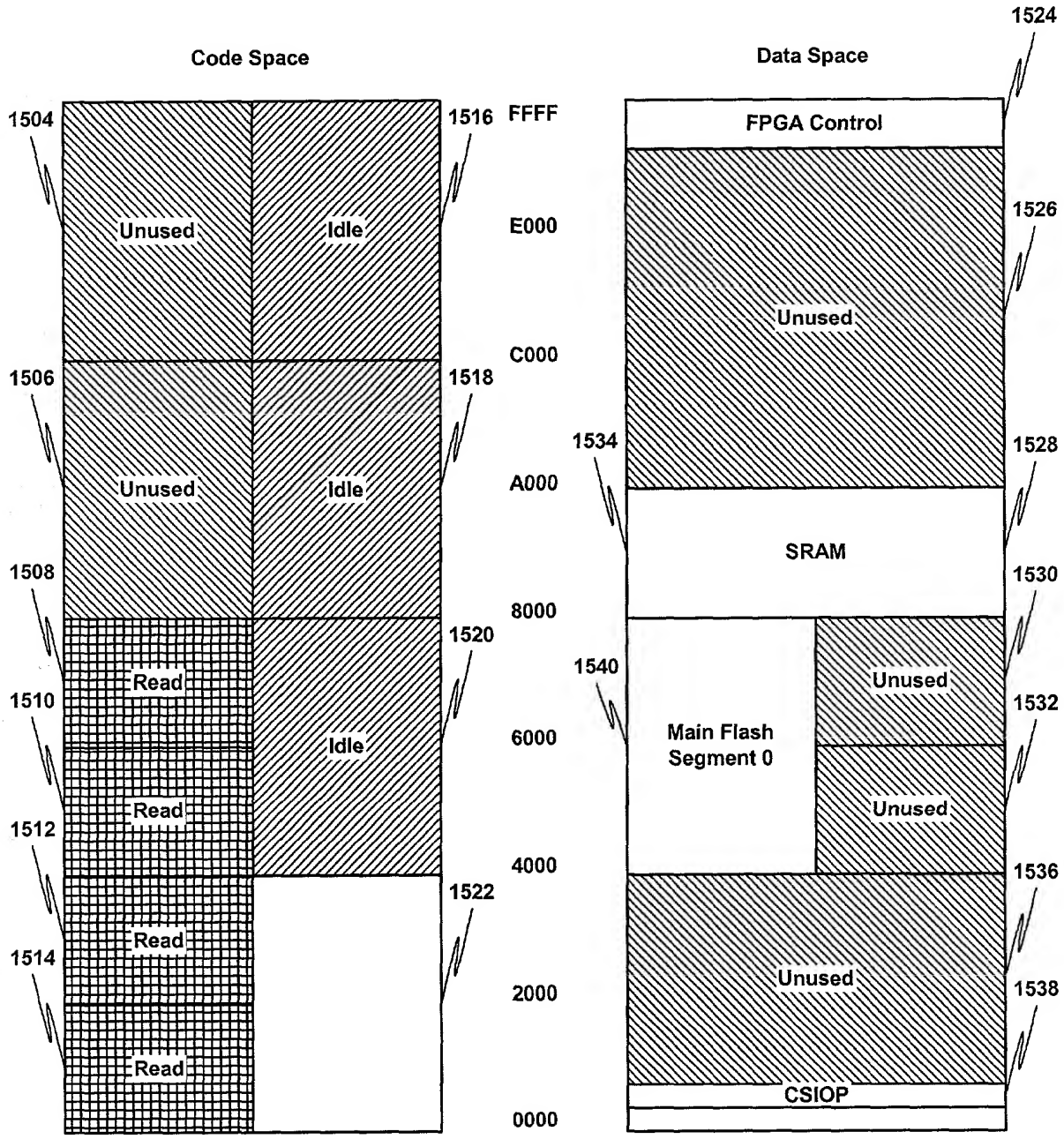

Code Space
Data Space
1504
1516
FFFF
Unused
Idle
E000
1524
FPGA Control
1526
Unused
C000
1506
1518
Unused
Idle
1534
1528
A000
SRAM
1508
1530
8000
1520
Read
1540
Unused
1510
1532
Read
Idle
Main Flash
Segment 0
Unused
1512
6000
Read
4000
1536
1522
1514
Read
Unused
2000
1538
Read
CSIOP
0000

FIG. 15

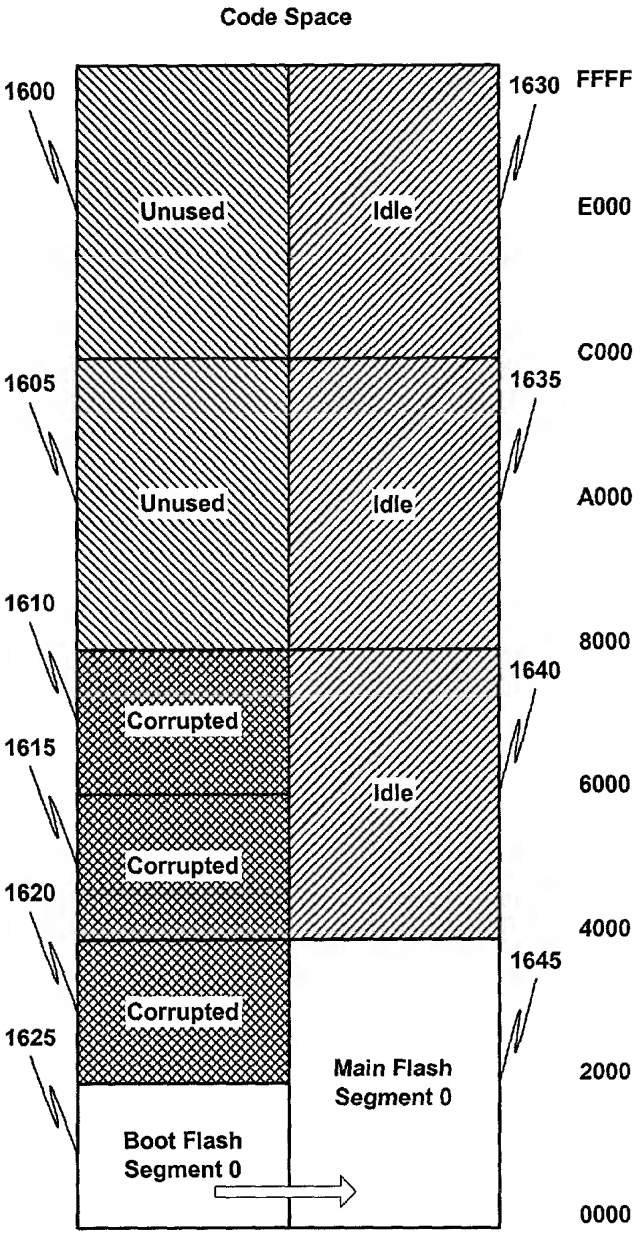

Code Space
1600
1630
FFFF
Unused
Idle
E000
C000
1605
1635
Unused
Idle
A000
1610
8000
1640
Corrupted
1615
Idle
6000
Corrupted
1620
4000
1645
Corrupted
1625
Main Flash
Segment 0
2000
Boot Flash
Segment 0
0000

FIG. 16

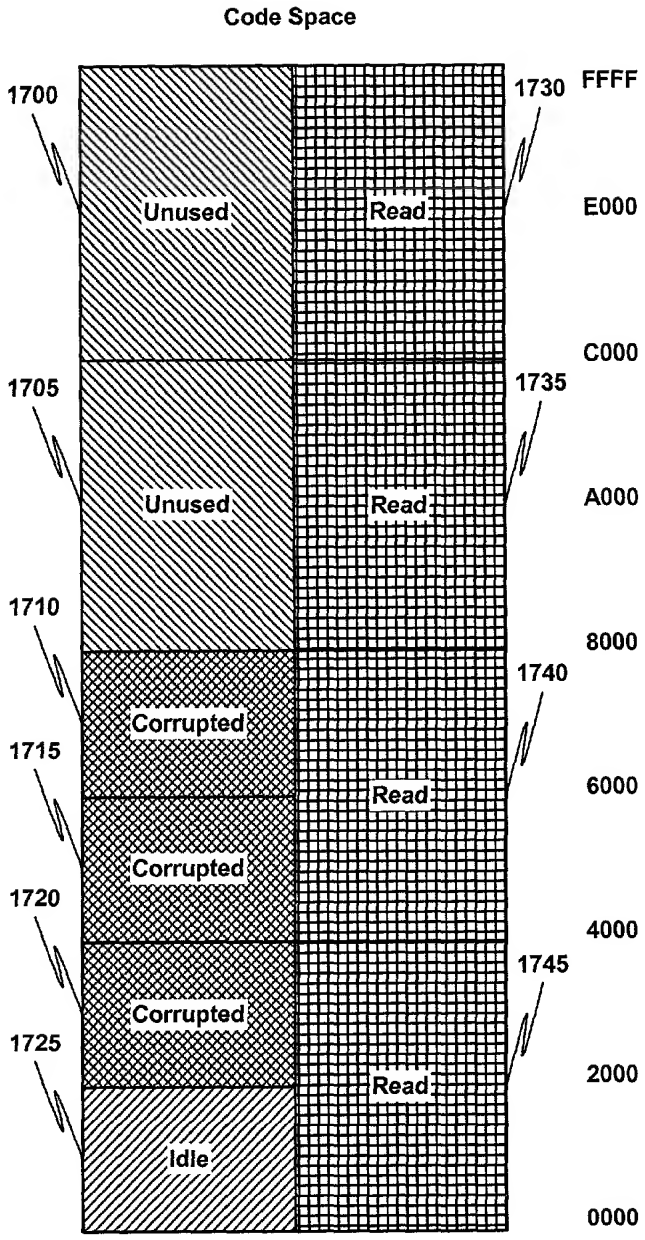

Code Space
1700
Unused
1705
Unused
1710
Corrupted
1715
Corrupted
1720
Corrupted
1725
Idle
1730
Read
1735
Read
1740
Read
1745
Read
FFFF
E000
C000
A000
8000
6000
4000
2000
0000

FIG. 17

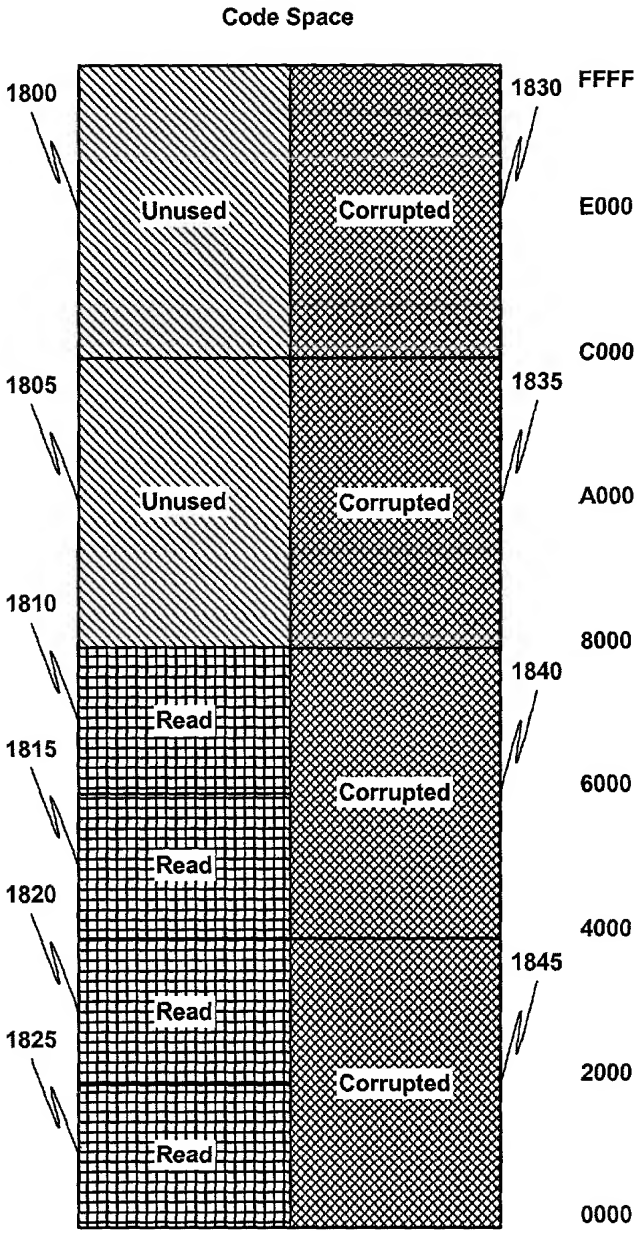

Code Space
1800
1805
1810
1815
1820
1825
1830
1835
1840
1845
Unused
Unused
Read
Read
Read
Read
Corrupted
Corrupted
Corrupted
Corrupted
FFFF
E000
C000
A000
8000
6000
4000
2000
0000

FIG. 18

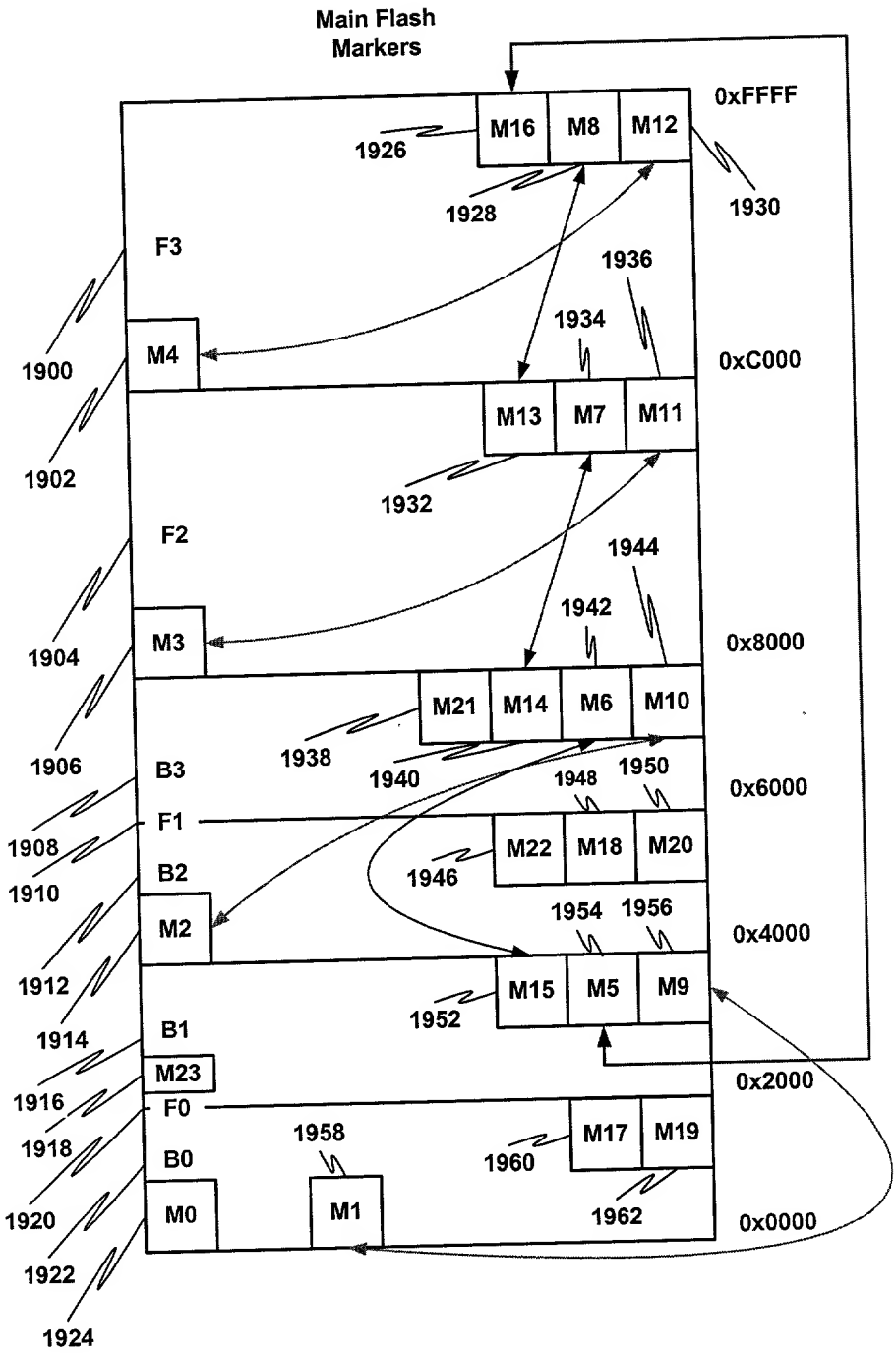

Main Flash Markers
0xFFFF
M16
M8
M12
1926
1928
1930
F3
1936
1934
M4
0xC000
1900
1902
M13
M7
M11
1932
F2
1944
1942
M3
0x8000
1904
1906
M21
M14
M6
M10
1938
1940
1948
1950
B3
0x6000
F1
1908
1910
M22
M18
M20
1946
B2
M2
1954
1956
0x4000
1912
1914
M15
M5
M9
1952
B1
M23
0x2000
1916
1918
F0
1958
M17
M19
1960
B0
M0
M1
1962
0x0000
1920
1922
1924

FIG. 19

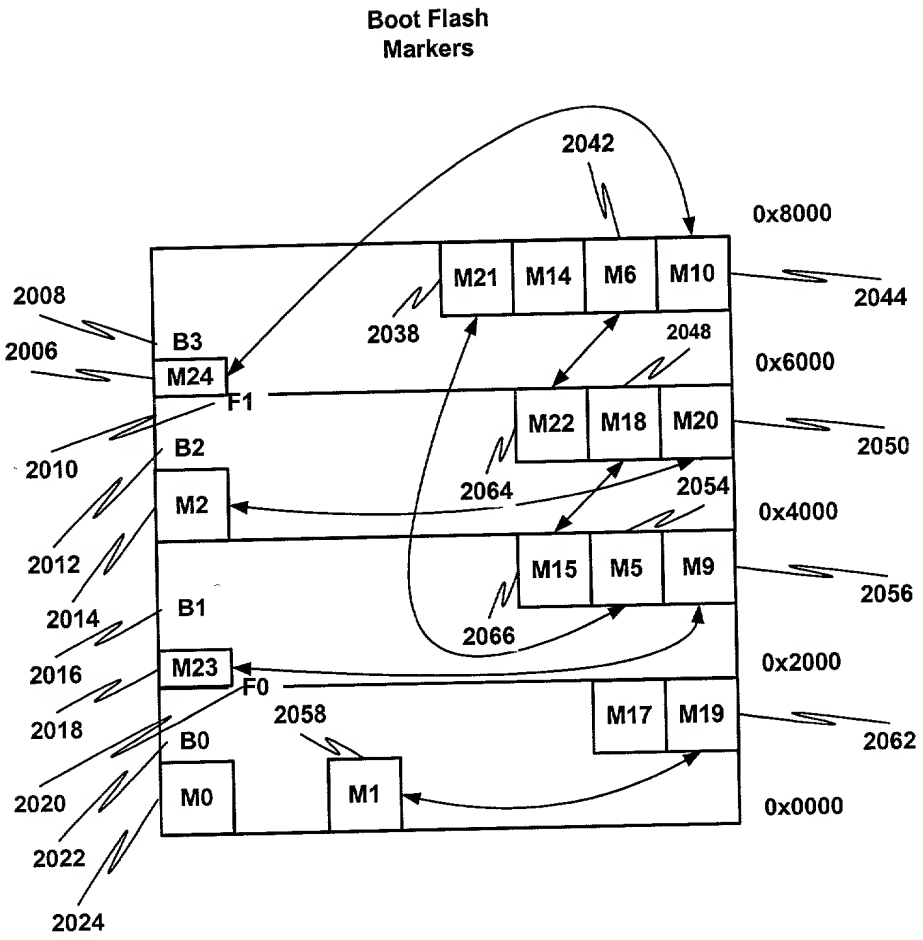

Boot Flash
Markers
2042
0x8000
M21
M14
M6
M10
2044
2008
2038
2048
2006
B3
M24
F1
0x6000
M22
M18
M20
2050
2010
B2
2064
2054
M2
0x4000
2012
M15
M5
M9
2056
2014
B1
2066
2016
M23
0x2000
F0
2018
2058
M17
M19
2062
B0
2020
M0
M1
0x0000
2022
2024

FIG. 20

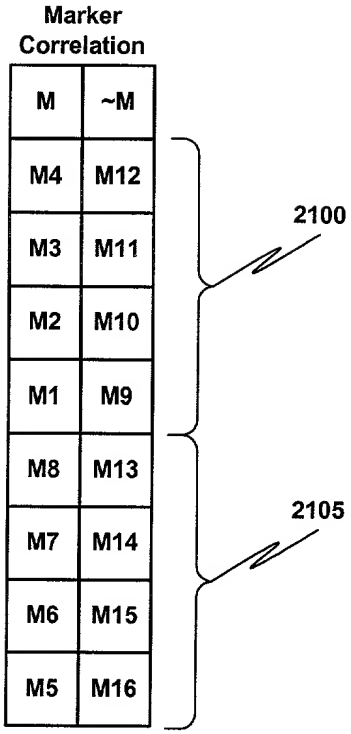


FIG. 21A

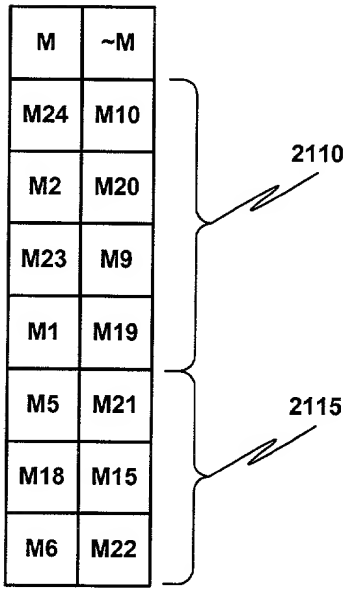


FIG. 21B

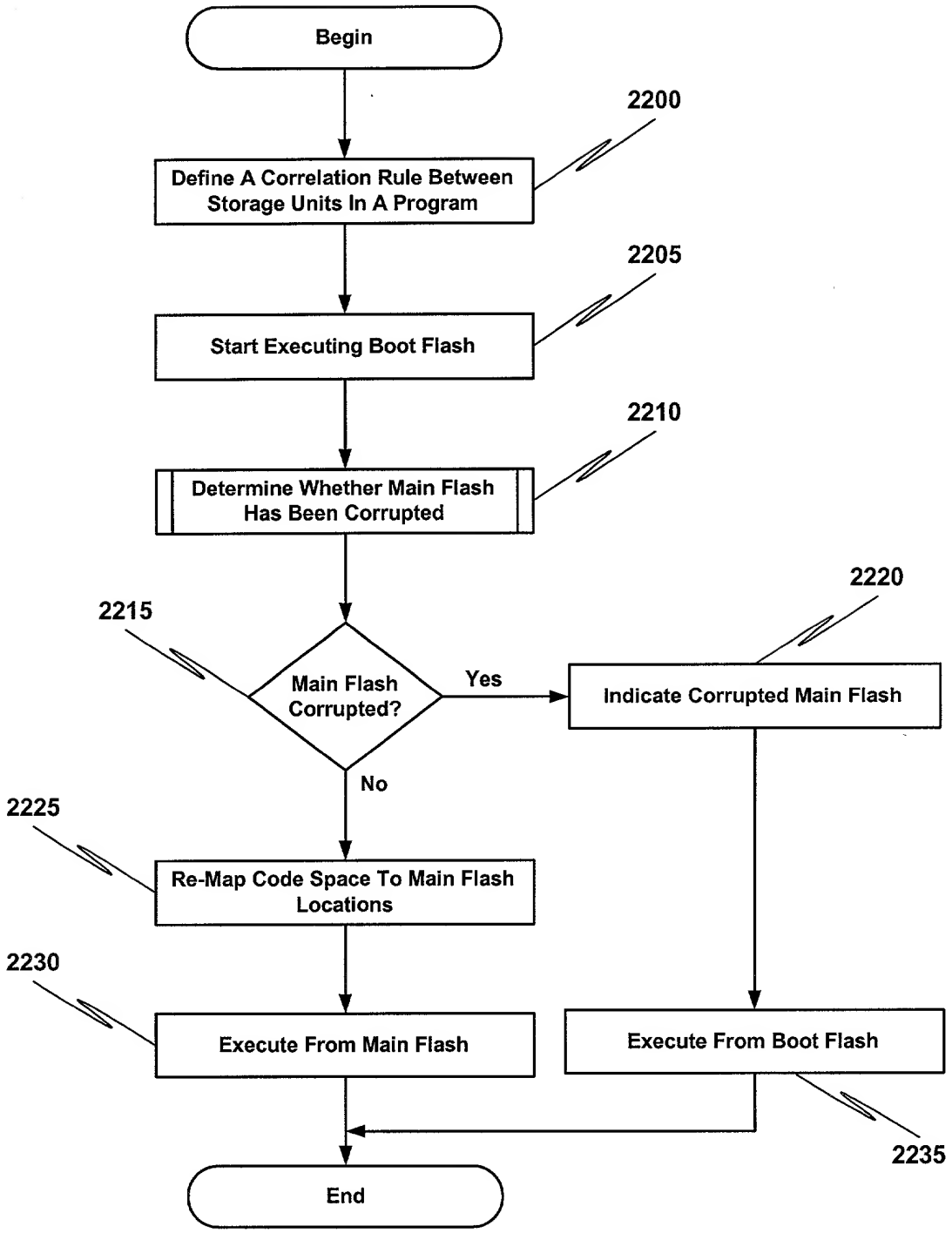

Begin
2200
Define A Correlation Rule Between Storage Units In A Program
2205
Start Executing Boot Flash
2210
Determine Whether Main Flash Has Been Corrupted
2215
Main Flash Corrupted?
Yes
2220
Indicate Corrupted Main Flash
No
2225
Re-Map Code Space To Main Flash Locations
2230
Execute From Main Flash
Execute From Boot Flash
2235
End

FIG. 22

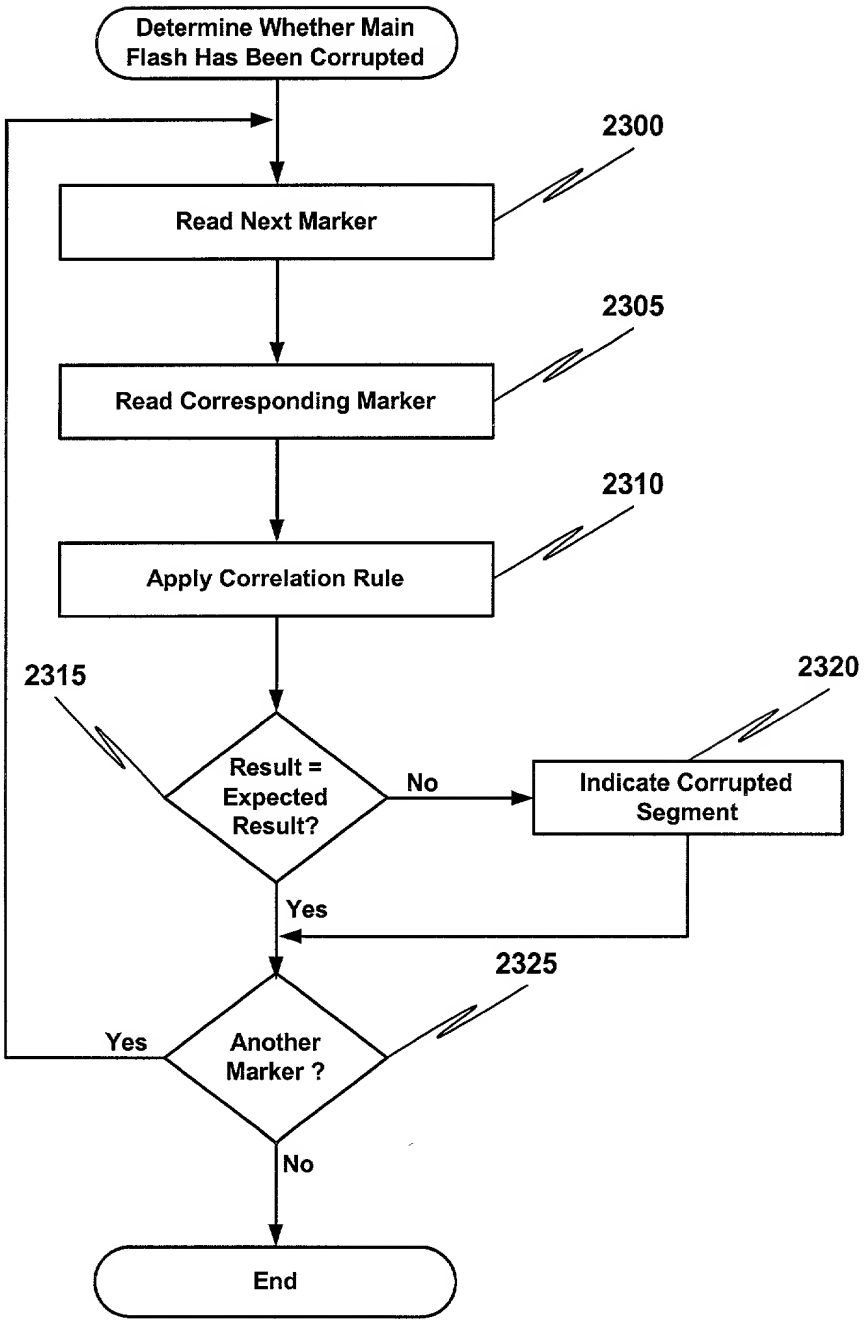

Determine Whether Main Flash Has Been Corrupted
2300
Read Next Marker
2305
Read Corresponding Marker
2310
Apply Correlation Rule
2315
Result = Expected Result?
No
2320
Indicate Corrupted Segment
Yes
2325
Another Marker ?
Yes
No
End

FIG. 23